NYC Taxi Fare Prediction

Ehsanur Rahman Rhythm
Department of Computer Science & Engineering
Brac University
66 Mohakhali,
Dhaka - 1212, Bangladesh
ehsanur.rahman.rhythm@g.bracu.ac.bd

Rajvir Ahmed Shuvo
Department of Computer Science & Engineering
Brac University
66 Mohakhali,
Dhaka - 1212, Bangladesh
rajvir.ahmed.shuvo@g.bracu.ac.bd

S.M.Azwad-Ul-Alam
Department of Computer Science & Engineering
Brac University
66 Mohakhali,
Dhaka - 1212, Bangladesh
s.m.azwad.ul.alam@g.bracu.ac.bd

***Abstract*—Archived information is used in predictive modeling to foretell the future. In light of significant developments in this area, predictive modeling has attracted a lot of interest in recent years. This is also used by businesses to improve the accuracy of your predictions, such as the amount you expect to pay for a taxi journey in the city. For the most part, cab trips in New York City are what make the city's traffic go. People in New York City take a lot of trips every day, so we can learn a lot about commute times, road congestion, and other issues from their experiences. Competition from app-based taxi vendors like Lyft, Curb, and Uber is increasing, so traditional taxi services need to lower their rates to remain competitive. Users may plan their travels more efficiently with the aid of cost estimates. Drivers may use it to choose the most profitable route, increasing their income. Even when popular taxi app-based vendor services implement spike charges, the openness regarding pricing and trip duration will assist in attracting consumers. The ultimate goal of this study is to analyze all possible trends and use the results to forecast fares. Therefore, we used real-time data provided by clients at the outset of a journey or throughout the booking process to estimate the ride's cost. This data-set included taxi pickup location id, pickup date-time, taxi drop-off location id, drop-off date and time, total trip distance, and the number of passengers commuting to the location data available from the official Taxi and Limousine Commission website. Appropriate regular or airport fares would be deduced by analyzing this data. After that, we tested out the XGBoost and LightGBM models to see which one could better predict actual outcomes and establish causal links between time-sensitive factors. Finally, by contrasting the two algorithms, we can conclude that XGBoost is much more accurate, while LightGBM is faster for predicting taxi fare prices.**

***Index Terms*—Predictive Modelling, XGboost, LightGBM, Data Analysis, Data Cleansing, Manhattan Distance, Average RMSE**

I. Introduction

Taxis are one of the main modes of transportation in every city. Taxi rides comprise the majority of the city's traffic. The data from the monthly total of millions of taxi rides can shed light on congestion, roadblocks, and other large-scale phenomena. It's crucial to be able to estimate how much a taxi ride will cost, as the passenger will want to budget accordingly. In recent years, various online services like Uber, Curb, and Lyft have sprung up to offer ride-sharing options in cities. They are already providing the estimated fares upfront so that the passengers can plan their budget for the trip. Therefore, it is advantageous for both taxis and customers if the estimated fare is displayed prior to the taxi trip, as this will encourage customers to use taxis during times when ridesharing services implement surge pricing.

New York City is one of the world's busiest cities. Every month, New York City residents and visitors take millions of rides. Due to the size and traffic of the city, it is an efficient way to travel within the city. NYC Yellow Taxi data provided by the New York City Taxi and Limousine Commission to forecast taxi fares (TLC) [1] was used here. We used the Forecast Model from Predictive Model to calculate the fare rate for NYC Taxi based on the distance traveled and the number of passengers. One subfield of "advanced analytics," known as "predictive analytics," is applied to the task of foreseeing potential outcomes. Utilizing methods from statistics, data mining, machine learning, and artificial intelligence, it examines both new and old information to forecast what might happen next [2].

In this paper, we try to predict the fare amount and show the next steps to make a successful prediction mechanism to be used as a tool, talk over the architecture and final output of each of these approaches. Finally, by contrasting the two algorithms, we conclude that XGBoost is superior to LightGBM when it comes to estimating taxi fares.

II. Literature Review

Many research groups attempted to predict taxi fare rates in New York City using a variety of factors. Using XGBoost and Multi-Layer Perceptron models, Poongodi et al. predicted taxi trip durations [3]. They factored in the pickup and drop-off locations, travel distance, departure time, and passenger count. Another group used 440 million trips to predict pickup density using Random Forest and KNN (K-Nearest Neighbors Regression) [4]. Daulton et al. factored in things like pick-up time, date, month, year, weather, and latitude/longitude. High levels of noise were produced in the model's output activity maps despite the increased amount of data. In 2018, Google Cloud and Coursera held a Playground Prediction Competition on Kaggle to determine if it was possible to estimate the total cost of a taxi ride in New York City from the given pickup and dropoff points (including any tolls).

III. Methodology

The steps of fare prediction began with the collection of raw data-set. Data analysis began with data reprocessing and data cleansing. Thereafter, the data were transformed into the appropriate form for the chosen data models. The data was split into groups for testing and training. Finally, we predicted the fare and tried to see the accuracy of the prediction with the real fare to validate the models.

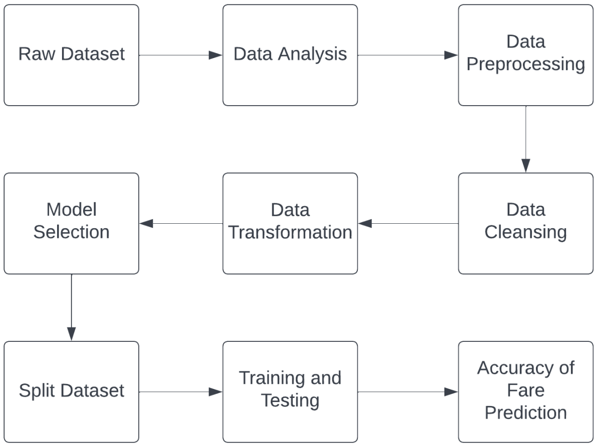


Fig. 1. Workflow of our Approach.

A. Description of Data

The dataset was sourced from the New York City Taxi and Limousine Commission. We used Yellow Taxi's trip data, which includes fields recording the date/time of pickup and drop-off, the location of pickup and drop-off, the total distance traveled, the total fare, the type of fare, the method of payment accepted, and the number of passengers the driver reported. The dataset was in PARQUET format, which was later converted into CSV format with the help of the panda's library in Python. For training the model, the dataset included the date, time, and location ID of pickup and drop-off, trip distance, total number of passengers, and fare. We took a whole month's worth of trip data, which included millions of rows. For training our model, we used half of the trip data from April 2022. The following is a concise description of each column in the dataset:

- **tpep_pickup_datetime:** The date and time the meter was first activated.
- **tpep_dropoff_datetime:** The date and time the meter was turned off.
- **PULocationID:** TLC Taxi Zone where the taximeter was activated.
- **DOLocationID:** TLC Taxi Zone where the taximeter was turned off.
- **trip_distance:** The distance traveled in miles as indicated by the taximeter.
- **passenger_count:** The number of passengers in the vehicle, as entered by the driver.
- **fare_amount:** The fare is based on time and distance as determined by the meter.

Since the taxi zones clustered the trip data, longitude and latitude were not used because it would require more computational power and more test data to produce an accurate result.

B. Analysis of Data

We first try to assign correlation among the columns of data by figuring out a heatmap, with a scale between 0 and 1. We do this by dropping the fare amount and visualizing the data set in the Pearson Correlation Heatmap.

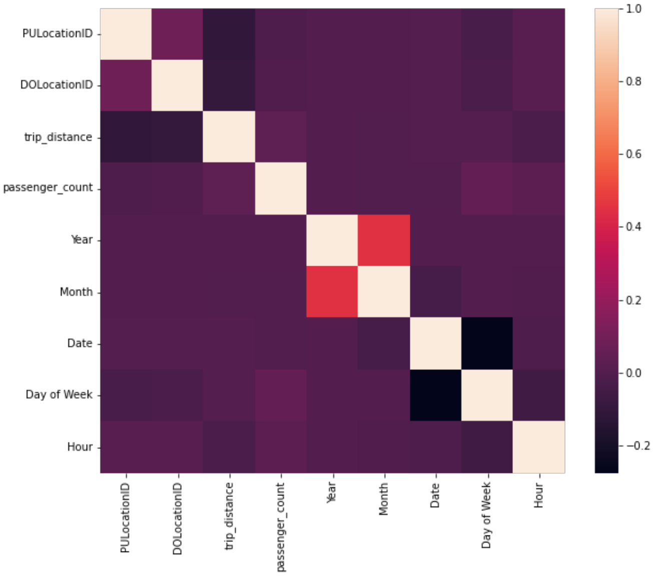


Fig. 2. Pearson Correlation Heatmap.

Next, we plot the trip distance data to see the frequency of the travel distance from the start. This is done in the distance distribution graph by using norm.fit() method. For a real number x, the given function of probability density for this method is:

$$f(x) = \frac{exp(-x^2/2)}{\sqrt{2\pi}}$$

This is represented in standardized form [5]. Additionally, we show the probability plot of the trip distance.

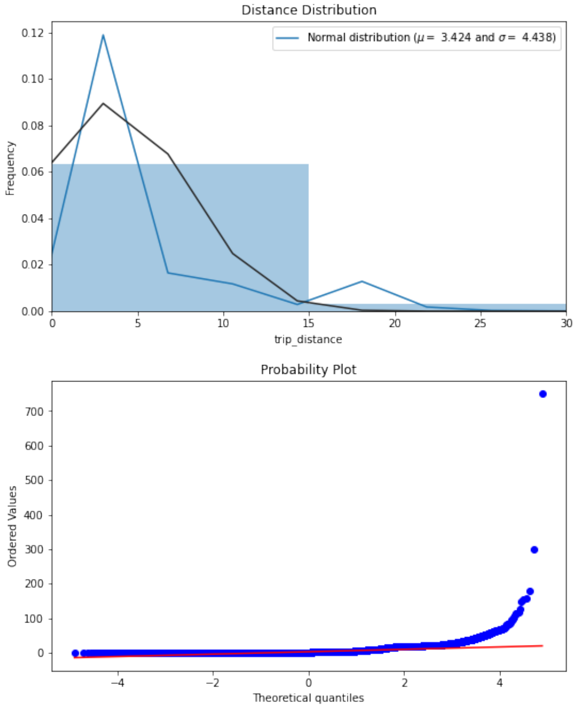


Fig. 3. Trip Distance Frequency and Probability Plot.

Next up, we implement the same norm.fit() method and probability plot to get the frequency distribution and probability plot for fare amount. This helps us locate what most people expect to pay on average in NYC.

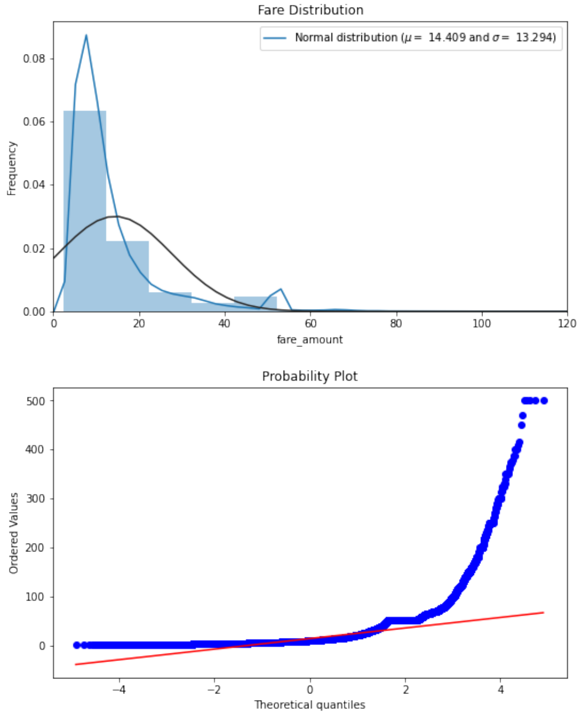


Fig. 4. Fare Amount Frequency and Probability Plot.

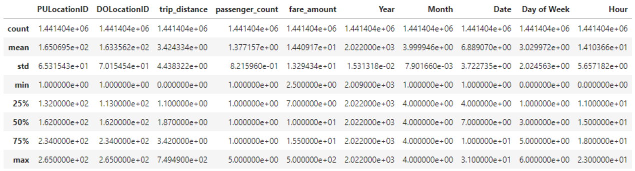
IV. Implementation

A. Data Pre-Processing

a) Obtaining Data: The data set was obtained from NYC Taxi and Limousine Commission (A government website) [1]. Only a subset of the whole data set from April 2022 (3.5 million taxi trips) was used due to computational and memory limitations. Furthermore, the first 1,500,000 rows of data were used to train the models. we used an 80% / 20% split for training and testing respectively. The validation set has 50 randomly chosen fares for comparison from March 2022.

b) Data Cleansing: As the data was in Apache parquet (column-based) format, it was transformed into CSV using the pandas library. Only 6 columns of data were retained from 19 columns for training purposes as they were our prime focus of discussion. Any passenger count below 5 was dropped and only the fair amount between 2.5 dollars to 500 dollars was kept from the dataset.

c) Data Transformation: The pick and drop-off date-time columns were divided into the columns year, month, date, date of week, and hour to see the correlation among multiple time-frames. We then drop all the null values and apply to describe() which returns generative statistics of the data.



	PULocationID	DOLocationID	trip_distance	passenger_count	fare_amount	Year	Month	Date	Day of Week	Hour
count	1.441404e+06	1.441404e+06	1.441404e+06	1.441404e+06	1.441404e+06	1.441404e+06	1.441404e+06	1.441404e+06	1.441404e+06	1.441404e+06
mean	1.650695e+02	1.633562e+02	3.424334e+00	1.377157e+00	1.440917e+01	2.022000e+03	3.999946e+00	6.889070e+00	3.029972e+00	1.410366e+01
std	6.531543e+01	7.015454e+01	4.438322e+00	8.215960e-01	1.329434e+01	1.531318e-02	7.901660e-03	3.722735e+00	2.024563e+00	5.657182e+00
min	1.000000e+00	1.000000e+00	0.000000e+00	1.000000e+00	2.500000e+00	2.009000e+03	1.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00
25%	1.320000e+02	1.130000e+02	1.100000e+00	1.000000e+00	7.000000e+00	2.022000e+03	4.000000e+00	4.000000e+00	1.000000e+00	1.100000e+01
50%	1.620000e+02	1.620000e+02	1.870000e+00	1.000000e+00	1.000000e+01	2.022000e+03	4.000000e+00	7.000000e+00	3.000000e+00	1.500000e+01
75%	2.340000e+02	2.340000e+02	3.420000e+00	1.000000e+00	1.550000e+01	2.022000e+03	4.000000e+00	1.000000e+01	5.000000e+00	1.800000e+01
max	2.650000e+02	2.650000e+02	7.494900e+02	5.000000e+00	5.000000e+02	2.022000e+03	4.000000e+00	3.100000e+01	6.000000e+00	2.300000e+01

Fig. 5. describe() method implanted on data.

Then we optimize the integer and float type data into a numeric type-data, so we can compare and contrast.

B. Models

a) XGBoost: Extreme gradient boosting, or XGBoost for short is a type of ensemble learning algorithm. It's a versatile kind of implementation that fully embraces the ideas of decision trees [6]. In addition, its speed is much improved in comparison to widely used algorithms like Adaboost. It has also lately become the most popular machine learning technique, garnering a lot of attention in Kaggle contests. The two most important considerations for us while using this algorithm are its speed of execution and its performance.

b) LightGBM: Positioning, classification, and many other AI tasks may all benefit from LightGBM, which is a fast, distributed, elite gradient boosting method based on decision tree computation. Unlike other boosting algorithms, which divide trees by their depth or level rather than their leaves, this one does so according to the best fit at each leaf. LightGBM's leaf-wise approach, which is based on the same tree, may decrease more loss than the level-wise strategy [7], leading to much higher accuracy. This is something that can only be done seldom by any of the current boosting techniques. And, as the name ”Light” suggests, it is very quick.

V. Results and Analysis

We used two models in this project. One is LightGBM and another is XGBoost Classifier. The r2 score and RMSE of XGBoost were better than LightGBM. When assessing the effectiveness of a machine learning model based on regression, the r2 score is a crucial metric. It measures the amount of variance in the predictions that the dataset can account for. On the other hand, when evaluating a model's performance, whether during training, cross-validation, or monitoring after deployment, RMSE is a very useful single metric to know. RMSE reveals the degree to which the data are concentrated around the line of best fit. The train r2 score of the XGBoost Classifier is 0.9154 whereas the train r2 score of LightGBM is 0.8768. Again, the test r2 score of the XGBoost Classifier is 0.8804 and the test r2 score of LightGBM is 0.8718. Now, the r2 score of XGBoost Classifier is better than that of LightGBM both in test and train. But if we compare the train and test r2 scores, the change is negligible in case of LightGBM which indicates LightGBM generalizes better than XGBoost Classifier. Again, in the case of RMSE we see that XGBoost has better score than LightGBM. The train XGBoost RMSE is 3.6714 and the test RMSE is 4.5674. So here we see the difference is quite much for RMSE which is happening due to the overfitting of the data in XGBoost. On the other hand, the train LightGBM RMSE is 4.3712 and the test RMSE is 4.7290. The two values are quite close so we can say that the LightGBM model is more consistent in testing the data.

TABLE I
r2 Score and RMSE Score of both models

Model	XGBoost	LightGBM
Train r2 score	0.92	0.88
Test r2 score	0.88	0.87
Train RMSE	3.67	4.37
Test RMSE	4.57	4.73

We boosted both the models and tested them against a test dataset that has 50 random trips from March 2022 in contrary to the training dataset being from April 2022. The XGBoost Classifier model places us in the 50th percentile in the Kaggle leaderboard but the LightGBM model is in the 43rd percentile in the Kaggle leaderboard. The scores might not be impressive but we have to consider the amount of data that we used. In the Kaggle competition, the dataset had data for the whole year of 2016 which had 50 million rows of data whereas we used the data for only April 2022. And not even the whole data of April 2022 we used half of the data of April 2022 which is almost 3% of the data that was used in the Kaggle Competition. We had limitations in terms of GPU, so our whole test was run by CPU which resulted in us in taking much longer time than usual.

We used 1.5 million rows of data for our whole project. We trained our model with 80% of these 1.5 million rows and 700 boost rounds in both LightGBM and XGBoost. We noticed a significant difference in time taken by the two models. LightGBM was almost 7 times faster than XGBoost in a Core i5 10th gen processor with 32GB RAM 3.10 GHz without any dedicated GPU. The accuracy of the models was not found as it is not relevant in this project as it not a classification model. Thus, we can conclude that in terms of scores XGBoost was better than LightGBM but LightGBM showed much more consistency with the results and was 7 times faster than XGBoost. If we could use more data, we could have verified it much more clearly as to which model is better for this project.

TABLE II
Predicted Fare (in USD) : XGBoost and LightGBM

Actual Fare	XGBoost	LightGBM
$10.00	$9.92	$10.19
$10.50	$9.16	$9.32
$52.00	$51.90	$52.22
$11.00	$12.59	$11.83
$25.00	$27.21	$25.91
$43.50	$42.69	$43.25
$4.50	$4.70	$4.40
$13.00	$11.36	$10.93
-$2.50	$6.77	$14.75

VI. Future Scope

We intend to test the data with another model, CatBoost, and compare the three models. Because of the lack of GPU power, we had to do the computing with CPU and we would like to use the whole GPU power and something on the level of 7000 Boost rounds to test the whole 50 million rows of data. Furthermore, our dream is to use this model practically in Bangladesh in the case of taxi cabs, buses, CNG, or even rickshaws. We would also love to implement real-time traffic data to the whole model, to predict fare of these vehicles relative to the traffic jams that we so often in Bangladesh.

VII. Conclusion

Predicting any kind of future with high accuracy is challenging nonetheless. Moreover, with various kinds of taxi fare data constantly changing and updating, data flow can be hard to handle especially by canceling out the noise. But by implementing XGBoost and LightGBM and getting a fairly accurate result, we can closely approximate the taxi fare in NYC. With RMSE score of XGBoost(train=3.6714, test=4.5674) < RMSE score of LightGBM(train= 4.3712, test=4.7290), indicating that XGBoost is much more accurate. But data consistency and faster implementation time become an issue, LightGBM is the preferred model in this regard. Nonetheless, as the cost of predictive modeling technology is rapidly decreasing in the current environment, we hope to conquer the challenges regarding compatibility and accurate prediction with existing and future models; given that conflicts are handled in a proper way.

Acknowledgment

First of all, we want to thank the Almighty Allah for His blessings. Additionally, we want to thank our course instructor for the course CSE438 from BRAC University, Ms.Sifat E

Jahan, as she guided us through the project and gave ample instructions to totally understand the work at hand. She has been pushing us to make our work better at every step.

References

[1] 2022. [Online]. Available: https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

[2] C. Elkan, *Predictive analytics and data mining*. University of California San Diego, 2013, vol. 600.

[3] M. Poongodi, M. Malviya, C. Kumar, M. Hamdi, V. Vijayakumar, J. Nebhen, and H. Alyamani, "New york city taxi trip duration prediction using MLP and XGBoost," *International Journal of System Assurance Engineering and Management*, vol. 13, no. S1, pp. 16–27, Jul. 2021. [Online]. Available: https://doi.org/10.1007/s13198-021-01130-x

[4] S. Daulton, S. Raman, and T. Kindt, "Nyc taxi data prediction," 2015. [Online]. Available: https://sdaulton.github.io/TaxiPrediction/.

[5] T. Kim, G. Lee, and B. D. Youn, "Uncertainty characterization under measurement errors using maximum likelihood estimation: cantilever beam end-to-end uq test problem," *Structural and Multidisciplinary Optimization*, vol. 59, pp. 1–11, 02 2019.

[6] A. Gupta, S. Sharma, S. Goyal, and M. Rashid, "Novel xgboost tuned machine learning model for software bug prediction," in *2020 International Conference on Intelligent Engineering and Management (ICIEM)*, 2020, pp. 376–380.

[7] D. Zhang and Y. Gong, "The comparison of lightgbm and xgboost coupling factor analysis and prediagnosis of acute liver failure," *IEEE Access*, vol. 8, pp. 220 990–221 003, 01 2020.